

## **REMARKS**

### **I. Introduction**

Claims 179 to 181 and 183 to 203 are currently pending in the present application, since claims 1 to 178, and 182 were previously canceled. In view of the foregoing amendments and the following remarks, it is respectfully submitted that all of the presently pending claims are allowable, and reconsideration of the present application is respectfully requested.

Applicants thank the Examiner for considering the previously filed Information Disclosure Statement, 1449 papers, and cited references.

### **II. Objection to the Specification**

The Office Action objects to the Specification as assertedly failing to include an Abstract that begins on a separate sheet in accordance with 37 C.F.R. § 1.52(b)(4). Applicants respectfully traverse this assertion. Applicants filed a Substitute Specification on June 25, 2002, which included an Abstract that commenced on a separate sheet – page 64. This Abstract was later amended by a replacement Abstract in Applicants' Response dated September 28, 2006.

The Office Action objects to the amendment in the Response dated September 28, 2006 because the amendment to the Abstract did not begin on a sheet separate from the amendments to other portions of the application. However, there is no requirement for amendments to the Abstract to begin a sheet separate from amendments to other portions of the Specification. As an initial matter, the amendment to the Abstract indicated that the Abstract previously submitted in the application as a whole beginning on a separate sheet (page 64) was to be replaced by the replacement Abstract included in Applicants' Response dated September 28, 2006, so that the Abstract included in the Response effectively now begins at page 64 of the originally filed application. That is, the replacement Abstract now begins on that separate sheet.

As for the manner of making amendments, the requirement of 37 C.F.R. § 1.52(b)(4) does not pertain to which sections of an amendment document must begin on a separate sheet. Instead, 37 C.F.R. § 1.121 sets forth the separate sections which must begin on separate sheets of the amendment document – amendment to the claims, amendment to the specification, replacement drawings, and remarks. Nowhere is there indicated to be a requirement for amendments to the Abstract to commence on a sheet separate from other amendments to the Specification.

Indeed, this is explicitly noted in M.P.E.P. § 714(II)(A)(B), which, in setting forth the sections which must begin on separate sheets, states “[t]he amendment papers should include, in the following order: . . . a section (must begin on a separate sheet) entitled ‘Amendments to the Specification’ (if there are any amendments to the specification). This section should include all amendments to the specification **including amendments to the abstract of the disclosure**” (emphasis added).

Accordingly, it is believed that no amendment is necessary.

### **III. Rejection of Claims 179 to 181 and 183 to 189 Under 35 U.S.C. § 102**

Claims 179 to 181 and 183 to 189 stand rejected under 35 U.S.C. § 102 as anticipated by U.S. Patent No. 5,801,958 (“the Dangelo reference”). It is respectfully submitted that the Dangelo reference does not anticipate any of claims 179 to 181 and 183 to 189, and the present rejection should be withdrawn, for at least the following reasons.

Claim 179 relates to a method for programming a system having a runtime reconfigurable cellular structure and recites, *inter alia*, the following:

*. . . separating the control flow graph into a plurality of subgraphs; distributing the plurality of subgraphs among a plurality of programmable hardware modules . . . ; determining state information for each of the subgraphs; and transferring the state information determined for one of the subgraphs from the one of the subgraphs to a subsequently executed subgraph.*

The Dangelo reference is unrelated to the subject matter of claim 179. The Dangelo reference does not refer to a system that has a configurable cellular structure and the programming thereof. Instead, the Dangelo reference refers to the theoretical design of a system and its representation in control and dataflow graphs to ultimately select an architecture which is to be built. The ultimately built architecture is not indicated to be a configurable cellular structure. Moreover, there is no discussion of programming the architecture that is ultimately built.

Notwithstanding the above, to facilitate matters, claim 179 has been amended herein without prejudice to further clarify the subject matter of the claim to explicitly recite that the structure is runtime reconfigurable and executes the program.

Moreover, with respect to state information and the transference of such information between subgraphs, the Office Action refers to partitions of the Dangelo reference as assertedly disclosing the subgraphs. While the Dangelo reference may refer generally to storing state information and retrieval of the information for display, nowhere does the Dangelo reference refer to transfer of state information from one partition to a subsequently executed partition (relied upon for disclosing the subgraphs). Thus, nowhere does the Dangelo reference disclose, or even suggest, transferring state information from one subgraph to a subsequently executed subgraph.

Thus, the Dangelo reference does not disclose, or even suggest, all of the features of claim 179, so that the Dangelo reference does not anticipate claim 179.

Claim 180 relates to a method for programming a system having a runtime reconfigurable cellular structure and recites, *inter alia*, the following:

*. . . extracting a data flow graph of a program that includes a loop; separating the data flow graph into a plurality of subgraphs, such that the loop is split into several of the subgraphs; and distributing the plurality of subgraphs among a plurality of hardware modules, such that the several subgraphs of the loop are distributed among at least two of the hardware modules.*

As noted above in support of the patentability of claim 179, the Dangelo reference is unrelated to the subject matter of claim 180. The Dangelo reference does not refer to a system that has a configurable cellular structure and the programming thereof. Instead, the Dangelo reference refers to the theoretical design of a system and its representation in control and dataflow graphs to ultimately select an architecture which is to be built. The ultimately built architecture is not indicated to be a configurable cellular structure. Moreover, there is no discussion of programming the architecture that is ultimately built.

Notwithstanding the above, to facilitate matters, claim 180 has been amended herein without prejudice to further clarify the subject matter of the claim to explicitly recite that the structure is runtime reconfigurable and executes the program.

Moreover, claim 180 provides for splitting of a loop into several subgraphs which are then distributed for execution. While the Dangelo reference generally refers to loops and generally refers to partitions, nowhere does the Dangelo reference refer to splitting any one loop, and certainly does not refer to distribution of parts of the split loop among different modules. For example, the Office Action makes reference to figure 36g of the Dangelo reference. Figure 36g merely shows an expansion of a block associated with a block of an else branch, which itself includes a while loop. It is not a portion split from a loop.

Thus, the Dangelo reference does not disclose, or even suggest, all of the features of claim 180, so that the Dangelo reference does not anticipate claim 180.

Claim 181, as herein amended without prejudice, relates to a method for programming a system having a cellular structure. As noted above in support of the patentability of claims 179 and 180, the Dangelo reference is unrelated to the subject matter of claim 181. The Dangelo reference does not refer to a system that has a configurable cellular structure and the programming thereof. Instead, the Dangelo reference refers to the theoretical design of a system and its representation in control and dataflow graphs to ultimately select an architecture which is to be built. The ultimately built architecture is not indicated to be a configurable cellular structure. Moreover, there is no discussion of programming the architecture that is ultimately built.

Notwithstanding the above, to facilitate matters, claim 181 has been amended herein without prejudice to further clarify the subject matter of the claim to explicitly recite that the structure is runtime reconfigurable and executes the program.

Thus, the Dangelo reference does not disclose, or even suggest, all of the features of claim 181, so that the Dangelo reference does not anticipate claim 181 or any of its dependent claims 183 to 189.

Withdrawal of this anticipation rejection of claims 179 to 181 and 183 to 189 is therefore respectfully requested.

#### **IV. Rejection of Claims 194 to 202 Under 35 U.S.C. § 102(b)**

Claims 194 to 202 stand rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 4,972,314 (“the Getzinger reference”). It is respectfully submitted that the Getzinger reference does not anticipate any of claims 194 to 202, and the present rejection should be withdrawn, for at least the following reasons.

Claim 194 is related to a method of executing a program on a runtime reconfigurable array of cells and recites, *inter alia*, “*after the computing, reconfiguring the first cell for computation of a first part of a second one of the subgraphs.*” The Office Action refers to the APs of the Getzinger reference as assertedly disclosing the cells of claim 194 that compute portions of graphs and are reconfigured between computations. However, nowhere does the Getzinger reference that the APs are reconfigurable and that any configuration of the APs is performed between processes of the APs.

Claim 194 further recites “*forming a plurality of subgraphs based on a program; computing a first part of a first one of the subgraphs with a first cell; . . . simultaneously with the reconfiguring, computing a second part of the first subgraph with a second cell.*” The Office Action refers to figure 5 and associated text of the Getzinger reference as assertedly disclosing this feature. Specifically, the Office Action asserts that Getzinger discloses that nodes A,C form a first subgraph and node B forms a second subgraph. The Office Action further asserts that Getzinger discloses that while one AP is configured to process node B, another AP processes subgraph A,C. However, the characterization of the Getzinger reference as disclosing these features is incorrect. Nowhere does the Getzinger reference indicate that nodes A,C form a subgraph defined separate from that of node B. Moreover, nowhere does the Getzinger reference indicate that node C is processed prior to processing of node B. To the contrary, the Getzinger reference indicates that node C is dependent on both A and B, so that both A and B would have to be processed prior to node C. Moreover, as explained above, the Getzinger reference does not indicate that the APs require reconfiguration. Instead, as soon as an AP is free from processing, it checks the queue to determine whether there is any node to be processed, and, if so, processes the node without any reconfiguration.

Claim 194 further recites that “*state information determined for one of the subgraphs is transferred from the one of the subgraphs to a subsequently executed subgraph.*” The Office Action refers to figure 47 of the Getzinger reference as assertedly disclosing the transfer of status information from one subgraph to another. However, the Office Action relied on the node structures of the type referred to in figure 5 as assertedly disclosing the subgraphs. The Office Action does not explain, nor is it readily apparent, how the status referred to in figure 47 relates to the nodes of figure 5. Indeed, at least to the extent the Office Action relies on the nodes of figure 5 as assertedly disclosing the subgraphs of claim 194, figure 47 does not disclose transfer of status information from one of the subgraphs to another.

For all of the foregoing reasons, the Getzinger reference does not disclose, or even suggest, all of the features of claim 194, so that the Getzinger reference does not anticipate claim 194 or any of its dependent claims 195 to 202.

**V. Rejection of Claims 190 to 193 Under 35 U.S.C. § 103(a)**

Claims 190 to 193 stand rejected under 35 U.S.C. § 103(a) as unpatentable over the combination of the Dangelo reference and U.S. Patent No. 6,421,808 (“the McGeer reference”). It is respectfully submitted that the combination of the Dangelo and McGeer references do not render unpatentable any of claims 190 to 193, and the present rejection should be withdrawn, for at least the following reasons.

Claim 190 relates to a method of executing a single program on a system having an array of runtime reconfigurable cells and recites, *inter alia*, the following:

*. . . separating the single program into several subgraphs; distributing the several subgraphs among different cells of the array; and executing the several subgraphs via the cells, the executing including: transmitting a data signal from a first cell via which a first one of the subgraphs is executed to a second cell via which a second one of the subgraphs is executed; and transmitting a status with the data signal, the status indicating whether the data signal is valid.*

As noted above in support of the patentability of claims 179 to 181, the Dangelo reference is unrelated to the subject matter of claim 190. The Dangelo reference does not refer to a system that has runtime reconfigurable cells and the programming thereof. Instead, the Dangelo reference refers to the theoretical design of a system and its representation in control and dataflow graphs to ultimately select an architecture which is to be built. The ultimately built architecture is not indicated to be a structure including runtime reconfigurable cells. Moreover, there is no discussion of programming the architecture that is ultimately built.

Moreover, claim 190 provides for separating a program into subgraphs distributed among the runtime reconfigurable cells for execution by the runtime reconfigurable cells. The Office Action refers to the partitions of the Dangelo reference as assertedly disclosing the subgraphs of claim 190. However, the partitions of the Dangelo reference are not distributed among an array of cells for execution by the cells. Instead, the partitions correspond to separate architectural sections of a hardware device which can ultimately be built by the conceptualization provided by the partitions.

The McGeer reference is not asserted to and does not correct these critical deficiencies of the Dangelo reference.

Accordingly, the combination of the Dangelo and McGeer references does not disclose or suggest all of the features of claim 190, so that the combination of the Dangelo and McGeer references does not render unpatentable claim 190 or any of its dependent claims 191 to 193.

Withdrawal of this obviousness rejection of claims 190 to 193 is therefore respectfully requested.

## **VI. Rejection of Claim 203 Under 35 U.S.C. § 103(a)**

Claim 203 stands rejected under 35 U.S.C. § 103(a) as unpatentable over the combination of the Dangelo reference and U.S. Patent No. 6,708,325 (“the Cooke reference”). It is respectfully submitted that the combination of the Dangelo and Cooke references does not render unpatentable claim 203, and the present rejection should be withdrawn, for at least the following reasons.

Claim 203 relates to a method for programming a system having a runtime configurable cellular structure, and, as herein amended without prejudice, recites, *inter alia*, the following:

*distributing the plurality of subgraphs among a plurality of hardware modules of the runtime configurable cellular structure; wherein: . . . the distribution of the plurality of subgraphs includes adapting the plurality of hardware modules such that state information determined for a first one of the subgraphs is transferred from the first one of the subgraphs to another subgraph that is to be subsequently executed.*

The Office Action refers to the Dangelo reference as assertedly disclosing these features. However, as noted above, the Dangelo reference is unrelated to the subject matter of claim 203. The Dangelo reference does not refer to a system that has runtime reconfigurable cells and the programming thereof. Instead, the Dangelo reference refers to the theoretical design of a system and its representation in control and dataflow graphs to ultimately select an architecture which is to be built. The ultimately built architecture is not indicated to be a structure including runtime reconfigurable cells. Moreover, there is no discussion of programming the architecture that is ultimately built.

Moreover, with respect to state information and the transference of such information between subgraphs, the Office Action refers to partitions of the Dangelo reference as assertedly disclosing the subgraphs. While the Dangelo reference may refer generally to storing state information and retrieval of the information for display, nowhere does the Dangelo reference refer to transfer of state information from one partition to a subsequently executed partition (relied upon for disclosing the subgraphs). Thus, nowhere does the Dangelo reference disclose, or even suggest, transferring state information from one subgraph to a subsequently executed subgraph.

Moreover, claim 203 recites that “*the extracting includes, for a conditional instruction of the program, extracting a plurality of different subgraphs, each representing a different instruction path of the conditional instruction.*” The Office Action refers to column 6, lines 23 to 67 of the Cooke reference as assertedly disclosing this feature. However, any review of the cited section makes plain that it does not disclose or suggest this feature. Instead, the cited section refers generally to partitioning of functions into FPGA blocks to minimize the required time for each separate FPGA block. Indeed, any review of the Cooke reference makes plain that it does not disclose or suggest extraction for a conditional instruction as provided for in the context of claim 203.

Moreover, claim 203 recites that “*for each one of the different subgraphs, the system sets execution of the subgraph to be bypassed as soon as an evaluation in accordance with the conditional instruction reveals that output of the subgraph will not be selected.*” The Office Action refers to column 8, lines 9 to 19 of the Cooke reference as assertedly disclosing this feature. However, the cited section merely refers to preloading of instructions based on branch prediction, which concerns a probable, and often faulty, branch selection. It does not concern a case of an evaluation that a subgraph output will not be selected, and does not concern bypass of any subgraph.

For all of the foregoing reasons, the combination of the Dangelo and Cooke references does not disclose or suggest all of the features of claim 203, so that the combination of the Dangelo and Cooke references does not render unpatentable claim 203.

Withdrawal of this obviousness rejection of claim 203 is therefore respectfully requested.

## **VII. Conclusion**

In light of the foregoing, it is respectfully submitted that all pending claims are in condition for allowance. Prompt reconsideration and allowance of the present application are therefore earnestly solicited.

Respectfully submitted,

Dated: July 22, 2009

By: /Aaron Grunberger/  
Aaron Grunberger  
Reg. No. 59,210

KENYON & KENYON LLP  
One Broadway  
New York, New York 10004  
(212) 425-7200  
**CUSTOMER NO. 26646**